Python Nodes Review

13 min

We have a few zany characters to keep track of and Python nodes may do just the trick. Let's get started...

Instructions

1.

Outside of `Node`, instantiate three nodes. None have an argument for `link_node`:

- the first has a value of `"likes to yak"` and be assigned to a variable `yacko`
- the second has a value of `"has a penchant for hoarding snacks"` and be assigned to `wacko`
- the third has a value of `"enjoys spending time in movie lots"` and be assigned to `dot`

2.

Now let's give these nodes some responsibilities! `yacko` can keep track of `dot` and `dot` can keep up with `wacko`. `wacko` can't keep track of anything though.

Below the newly created nodes, use your `.set_link_node()` method to give:

- `yacko` a `link_node` of `dot`
- `dot` a `link_node` of `wacko`

3.

Create two new variables, `dots_data`, and `wackos_data`. Use both getter methods to get `dot`'s value from `yacko` and get `wacko`'s value from `dot`.
Print `dots_data` and `wackos_data` to the console to see the results!

When your code is passing, take a moment to consider:

- How would you get `yacko`'s value?
- How could you get from `yacko` to `wacko`'s `value`?
- How do you think nodes could be helpful for keeping track of and storing information?

Hint

For example, if we had a node, `nurse`, that had a `link_node` of `doc`, we could get `doc`'s value like this:

```
nurse.get_link_node().get_value()
```

script.py

```
class Node:
  def __init__(self, value, link_node=None):
    self.value = value
    self.link_node = link_node

  def set_link_node(self, link_node):
    self.link_node = link_node

  def get_link_node(self):
    return self.link_node

  def get_value(self):
    return self.value

# Add your code below:
yacko = Node("likes to yak")
wacko = Node("has a penchant for hoarding snacks")
dot = Node("enjoys spending time in movie lots")

yacko.set_link_node(dot)
dot.set_link_node(wacko)

dots_data = yacko.get_link_node().get_value()
wackos_data = dot.get_link_node().get_value()

print(dots_data)
print(wackos_data)
```